

# Тема 4. Основные понятия языка Object Pascal

- Синтаксические диаграммы
- Состав языка
- Тип данных
- Арифметические и логические выражения

## 4.1. Синтаксические диаграммы

**Синтаксическая диаграмма** представляет собой графическое представление понятия языка с помощью базовых символов языка и ссылок на предыдущие диаграммы, соединенных стрелками.



- обозначение базового понятия



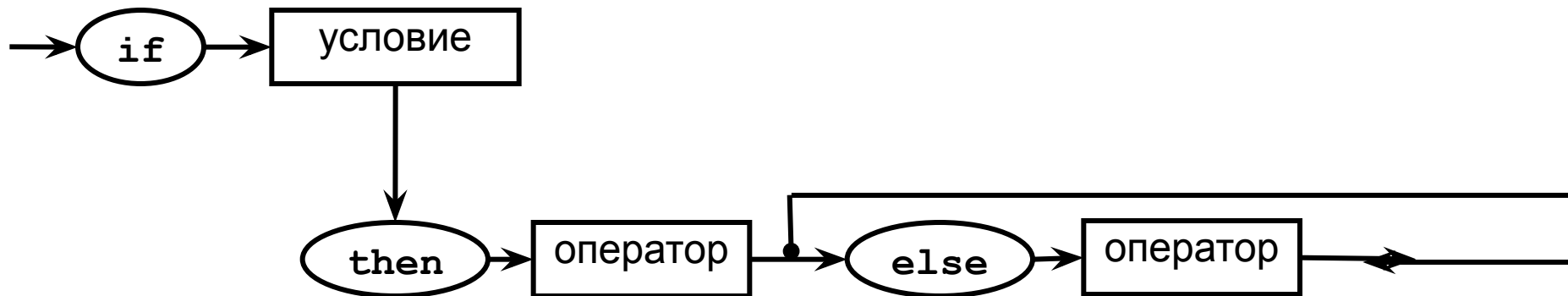
- обозначение понятия, расшифровывающегося диаграммой и использующегося для определения другого понятия



- обозначение соединения элементов диаграммы

# Пример синтаксической диаграммы

Условный оператор



**Сокращенная форма условного оператора:**

`if <условие> then <оператор>;`

**Полная форма условного оператора:**

`if <условие> then <оператор1> else <оператор2>;`

## 4.2. Состав языка

Любой язык определяется:

- **Алфавит** (набор символов, используемых в языке)
- **Синтаксис** (правила построения слов языка)
- **Семантика** (смысловое значение слов, выражений)

Алгоритмический язык определяется:

- **Алфавит** - набор неделимых символов
- **Лексема** - элементарные единицы языка, имеющие самостоятельный смысл
- **Выражение** – задает правило вычисления некоторого значения
- **Оператор** – задает описание некоторого действия
- **Директива компилятору** – задает процесс компиляции и построения программы

# Алфавит языка

- Латинские буквы

A...Z, a...z

- Арабские цифры

0 1 2 3 4 5 6 7 8 9

- Спецсимволы

+ - \* / = > < . , ; : \ @  
( ) [ ] { } \$ \_ # ^

символ пробела (в примерах обозначаем как `␣`)

- Кириллица (только для комментариев и строк)

A...Я, a...я

# Лексемы

служебные слова

знаки операций

разделители

идентификаторы

**Служебные слова** — это группа слов, смысл которых зафиксирован в языке:

```
Absolute  andarray  asm  assembler
Begin     case  Const  constructor  destructor
Div       do  downto  else  End
External  file  for  forward  function
Goto      if  implementation  in  inline
Interface Interrupt  label  mod
Nil       not object  of  or  packed
private   procedure  program  recored
Repeat    set shl  shr  string
Then      to  type  unit  until
Uses      var virtual  while  with  xor
```

**Знаки операций** – для задания действий по преобразованию данных и вычислению значений:

+   -   \*   /   =   <   <=   <>   >=   >

если используется двойной знак, то он набирается без пробела!

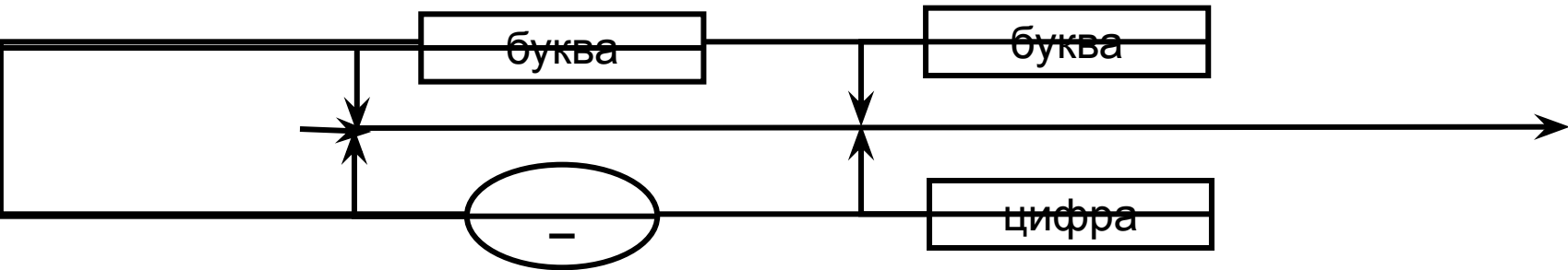
**Разделители:** «;» «:» «(» «)» «[», «]», «,» «.» « » (пробел)

**Пример:**     **Var**   a:integer;   b   :  real;c:   6

char;

**Идентификаторы** — это имена переменных, констант, процедур, функций, типов, меток, т.е. имя определенного программного объекта.

Идентификатор



**Примеры:** Summa      help1      count\_clock      q12345      \_assa  
S1      door      Mywindow

My\_Function  
my\_function  
MY\_FUNCTION

Не различаются!!!

Var l, i : integer;

Error: Identifier redeclared:

Служебные слова **НЕЛЬЗЯ** использовать в качестве имен переменных, определяемых пользователем.

**Примеры:** ~~Begin~~      MyBegin

~~Const~~      const1

## 4.3. Тип данных

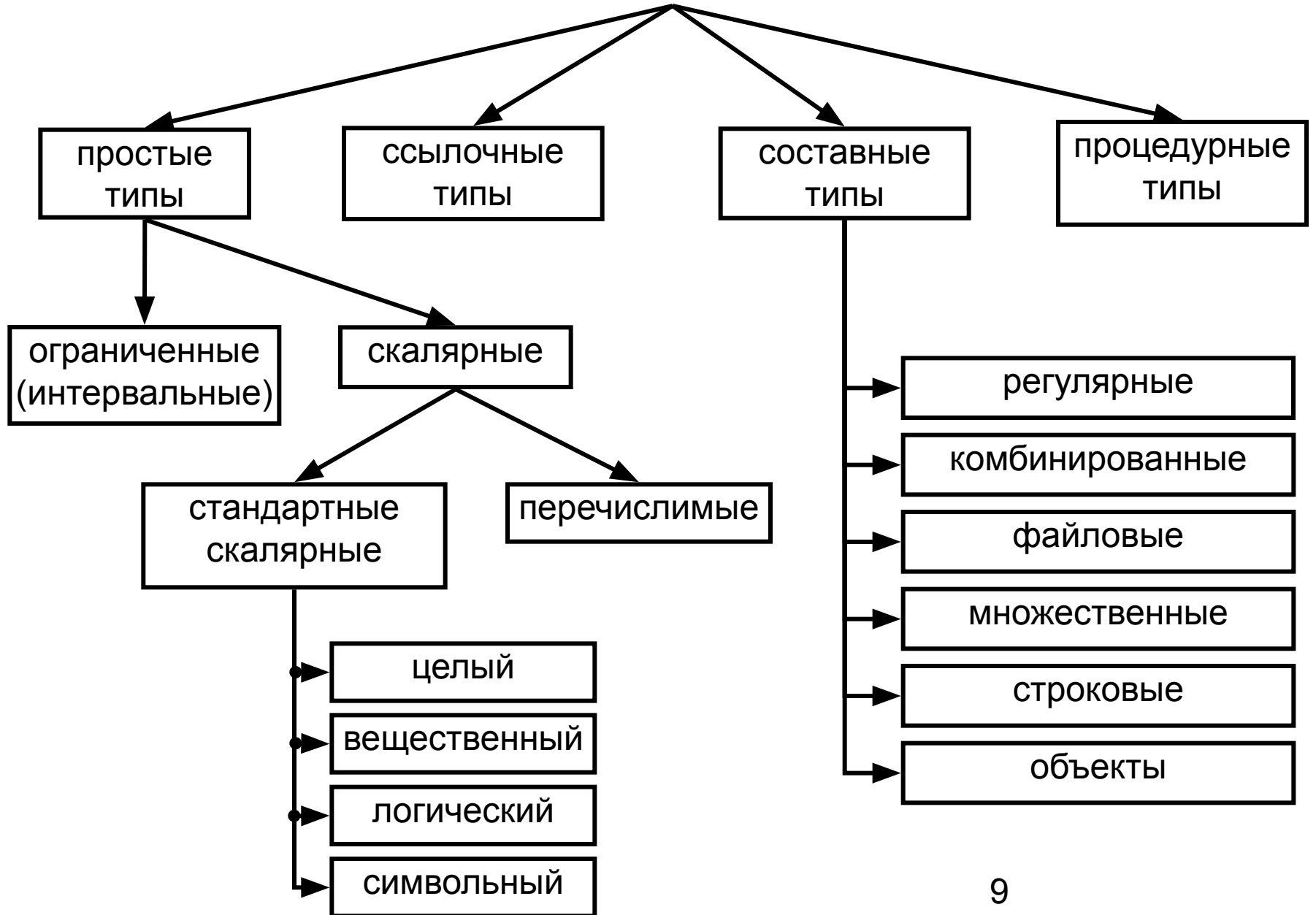
Маня

X

Тип определяет множество значений, допустимых для величин этого типа, и множество операций над НИМИ



## 4.3.1. Система типов языка Object Pascal



# стандартные скалярные типы данных

вещественный тип  
**Real**

целый тип  
**Integer**

символьный тип  
**Char**

логический тип  
**Boolean**

**дискретные типы**

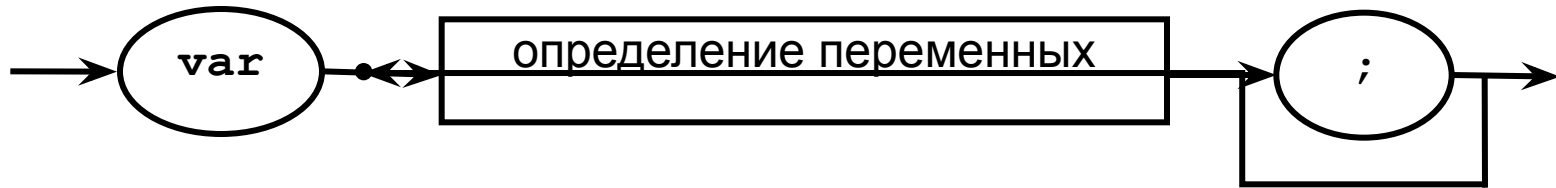
# Таблица простых типов

Название	Real	Integer	Boolean	Char
Диапазон значений	$5 \times 10^{-324} \dots 1,7 \times 10^{308}$	-2147483648 ... 2147483647	2	0...255
Занимаемая память	8 байт	4 байта	1 байт	1 байт
Операции	+ - * /	+ - * div mod	and or not xor	
	<	<= =	> >=	<>

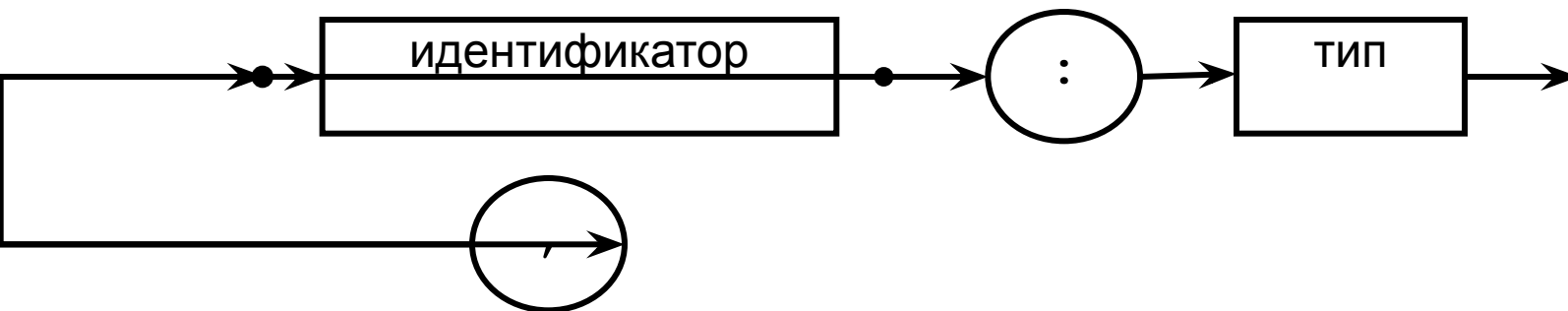
## 4.3.2. Описание данных

переменные и константы

Описание переменных



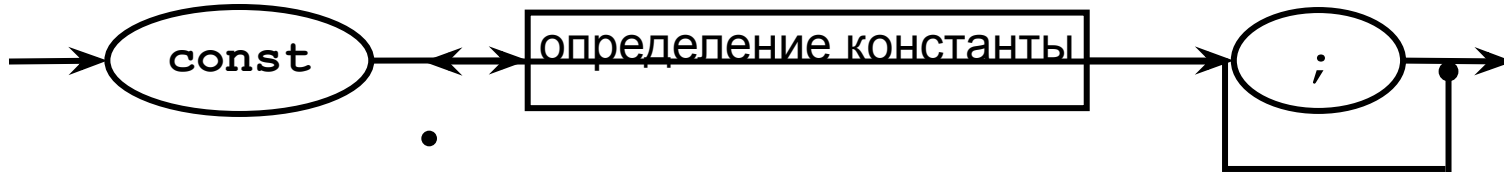
Определение переменных



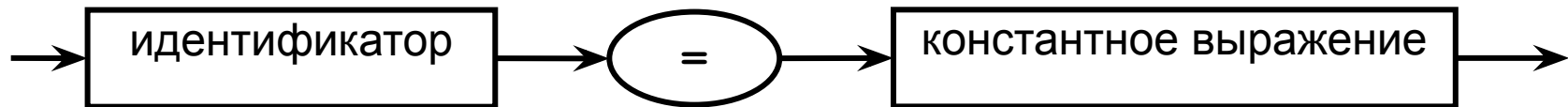
**Пример :**

```
Var i:integer; {описывается целая переменная i}  
      S:Real;   {описание дробной переменной S}
```

## Описание констант



## Определение константы



### Пример:

**Const** b=5;

*{это целая константа}*

g=9.34;

*{это дробная константа}*

c=' c ';

*{это символьная константа}*

MyConst=b+20;

*{это пример константного выражения}*

~~b:=25;~~

~~g:=g+100;~~

Значение констант в программе нельзя ~~В~~ менять

# 4.3.3. Стандартный логический тип Boolean

Описание констант:

```
Const MyBool = True;  
      MyFalse =  
False;
```

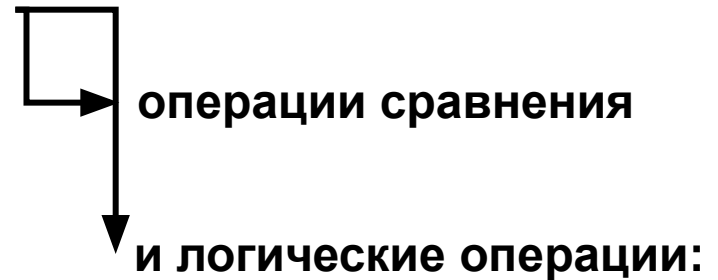
Описание переменных:

```
Var a,b: boolean;  
    bool: boolean;
```

Над логическими данными возможны только

~~True + False~~

True > False



A	Not A
1	0
0	1

A	B	A and B
1	1	1
1	0	0
0	1	0
0	0	0

A	B	A or B
1	1	1
1	0	1
0	1	1
0	0	0

A	B	A xor B
1	1	0
1	0	1
0	1	1
0	0	0

Результат операций – логический:

A < b  $\implies$  True или False

A or bool  $\implies$  True или False

# Дополнительные булевские типы

<b>Имя типа</b>	<b>Диапазон значений</b>	<b>Размер памяти</b>
ByteBool	2	1 байт
WordBool	2	2 байта
LongBool	2	4 байта

## 4.3.4. Стандартный целый тип **Integer**

Целое число



Описание констант:

```
Const myConst = 5;  
      Const1 = -27;  Const2 = $F1;
```

Описание переменных:

```
Var a,b: integer;
```

**Арифметические операции над целым типом:**

- сложение **+**, вычитание **-**, умножение **\***
- деление нацело (с отбрасыванием дробной части) **div**
- остаток от целочисленного деления **mod**

Результат операций  
– целого типа!

**Примеры:**

14 **div** 3 дает результат 4

14 **mod** 3 дает результат 2

14 **div** 2 дает результат 7

14 **mod** 2 дает результат 0

**Операции отношения над целым типом (результат - логический):**

$a = b$

$a \langle \rangle b$

$a > b$

$a \geq b$

$a < b$

$a \leq b$

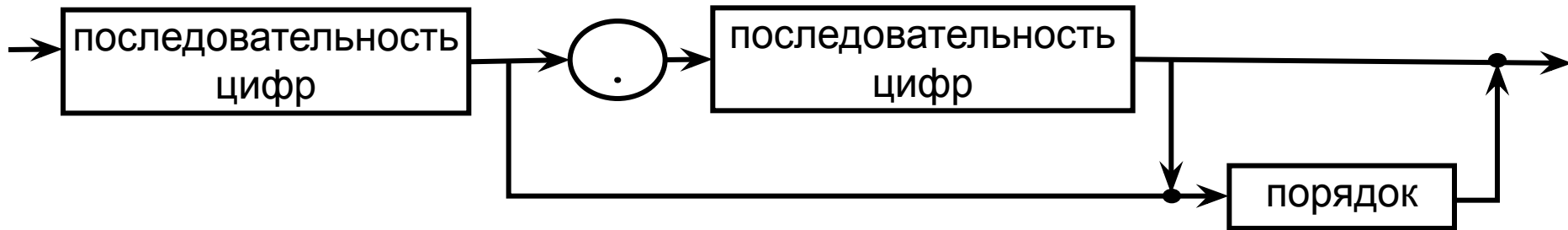


# Дополнительные целочисленные типы

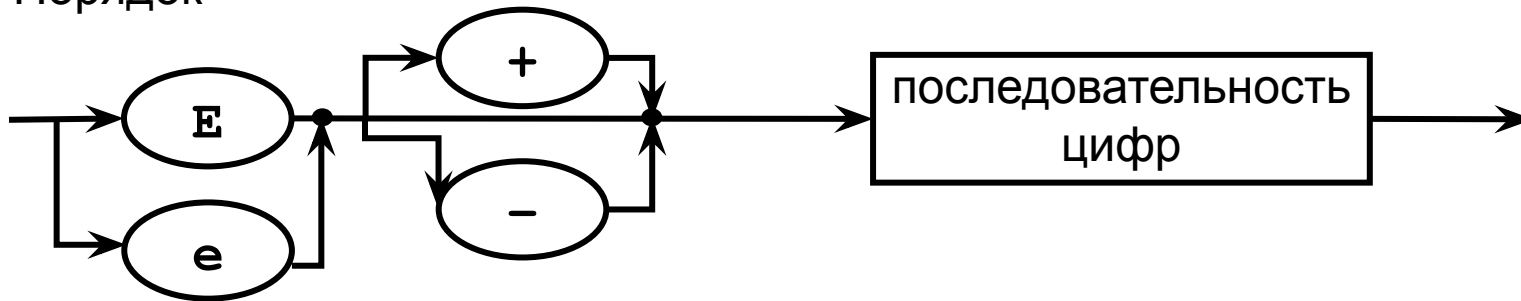
Имя типа	Диапазон значений	Размер памяти
byte	0 .. 255	1 байт
word	0 .. 65535	2 байта
longint	-2147483648 .. 2147483647	4 байта
shortint	-128 .. 127	1 байт
Cardinal	0..4294967295	4 байта
SmallInt	-32768..32767	2 байта
Int64	$-2^{63} .. 2^{63}-1$	8 байт
Longword	0..4294967295	4 байта

## 4.3.5. Стандартный вещественный тип **Real**

Вещественное число



Порядок



Последовательность цифр – одна или несколько арабских цифр, записанных подряд без пробела

## вещественные константы:

Форма с фиксированной точкой	Форма с плавающей точкой	мЕр мантисса (целое или вещественное число с фикс. т.) признак записи числа с порядком порядок числа (только целое число)
3.14	0.314E+01	
-7.8	-7.8E+00	
35.785	357.85E-01	

## Описание переменных:

**Var** a, b: real;

**Арифметические операции над вещественным типом:** + - \* /

Результат операций - вещественный

## Пример:

A+b **результат** число типа Real

**Операции отношения над вещественным типом:** = <> < > <=

Результат операций - логический

>=

## Пример:

a < b **результат** True или

False

# Дополнительные вещественные типы

Имя типа	Диапазон значений	Число цифр мантиссы	Размер памяти
single	1.5e-45 .. 3.4e38	7 - 8	4 байта
double	5.0e-324 .. 1.7e308	15 - 16	8 байт
extended	$\pm 3.6 \times 10^{-4951} \dots \pm 1.1 \times 10^{4932}$	19 - 20	10 байт
comp	$-2^{63}+1 \dots 2^{63}-1$	19 - 20	8 байт
Real48	$2.9 \times 10^{-39} \dots 1.7 \times 10^{38}$	11 - 12	6 байт
Currency	-922337203685477.5808 .. 922337203685477.5807	19 - 20	8 байт

## Пример:

```
Var r: real;  
    cc: Currency;
```

. . .

r имеет значение 5

cc имеет значение 5



```
C:\  
r= 5.0000000000000000E+0000  
cc= 5.0000000000000000E+0000
```

# 4.3.6. Стандартный символьный тип

## Char

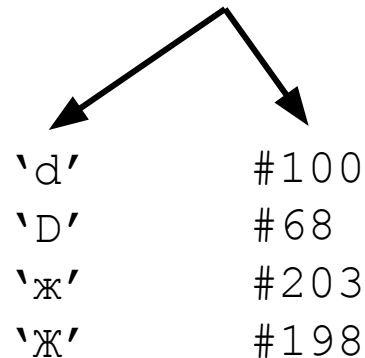
Коды 0...127  
(кодировка ASCII)

	000	016	032	048	064	080	096	112	
00		◆		0	@	P	`	p	00
01		◆	!	1	A	Q	a	q	01
02		↕	"	2	B	R	b	r	02
03	♥		#	3	C	S	c	s	03
04	◆	↑	\$	4	D	T	d	t	04
05	♣	§	%	5	E	U	e	u	05
06	♠		&	6	F	V	f	v	06
07	·		'	7	G	W	g	w	07
08		↑	(	8	H	X	h	x	08
09	°	↓	)	9	I	Y	i	y	09
10		→	*	:	J	Z	j	z	10
11		←	+	;	K	[	k	{	11
12		└	,	<	L	\	l		12
13		•	-	=	M	]	m	}	13
14		·	.	>	N	^	n	~	14
15	Ц	◆	/	?	O	_	o	\$	15

Коды 128...255  
(модифицированный  
альтернативный вариант)

	128	144	160	176	192	208	224	240	
00	А	Р	а	⊞	Л	Е	р	≡	00
01	Б	С	б	⊞	└	┘	с	±	01
02	В	Т	в	⊞	┘	└	т	≥	02
03	Г	У	г		┘	└	у	≤	03
04	Д	Ф	д		-	Е	ф		04
05	Е	Х	е	┘	┘	┘	х	┘	05
06	Ж	Ц	ж	┘	┘	┘	ц	÷	06
07	З	Ч	з	┘	┘	┘	ч	≈	07
08	И	Ш	и	┘	┘	┘	ш	°	08
09	Й	Щ	й	┘	┘	┘	щ	·	09
10	К	Ъ	к	┘	┘	┘	ъ	·	10
11	Л	Ы	л	┘	┘	┘	ы	√	11
12	М	Ь	м	┘	┘	┘	ь	°	12
13	Н	Э	н	┘	┘	┘	э	°	13
14	О	Ю	о	┘	┘	┘	ю	■	14
15	П	Я	п	┘	┘	┘	я		15

СИМВОЛЬНЫЕ КОНСТАНТЫ



СИМВОЛЬНЫЕ ПЕРЕМЕННЫЕ:

```
Var a,b: char;
    ch: char;
```

Символьные данные  
можно только  
сравнивать

Пример:

```
'a' > 'A' → False
'y' < 'Y' → True
```

# Дополнительные символьные типы

<b>Имя типа</b>	<b>Диапазон значений</b>	<b>Размер памяти</b>
<code>AnsiChar</code>	0 .. 255	1 байт
<code>WideChar</code>	0 .. 65535	2 байта

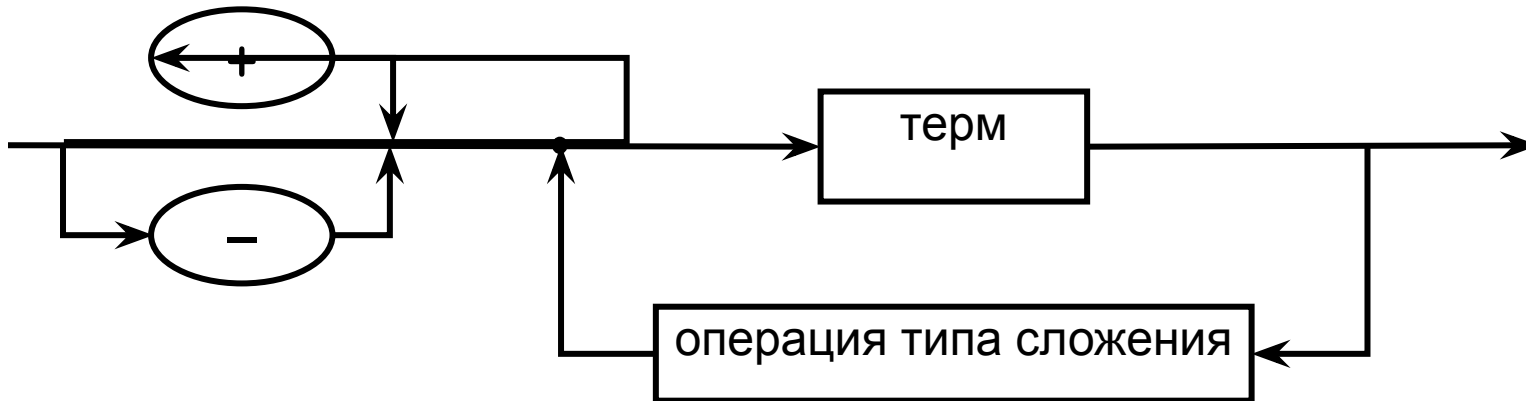
## 4.4. Арифметические и логические выражения

Выражение



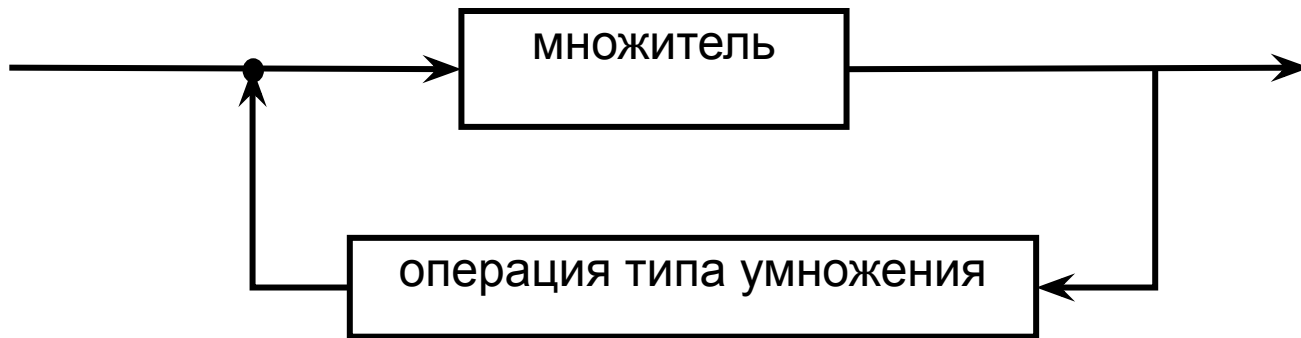
Операции отношения: = <> < > <= >=

Простое выражение



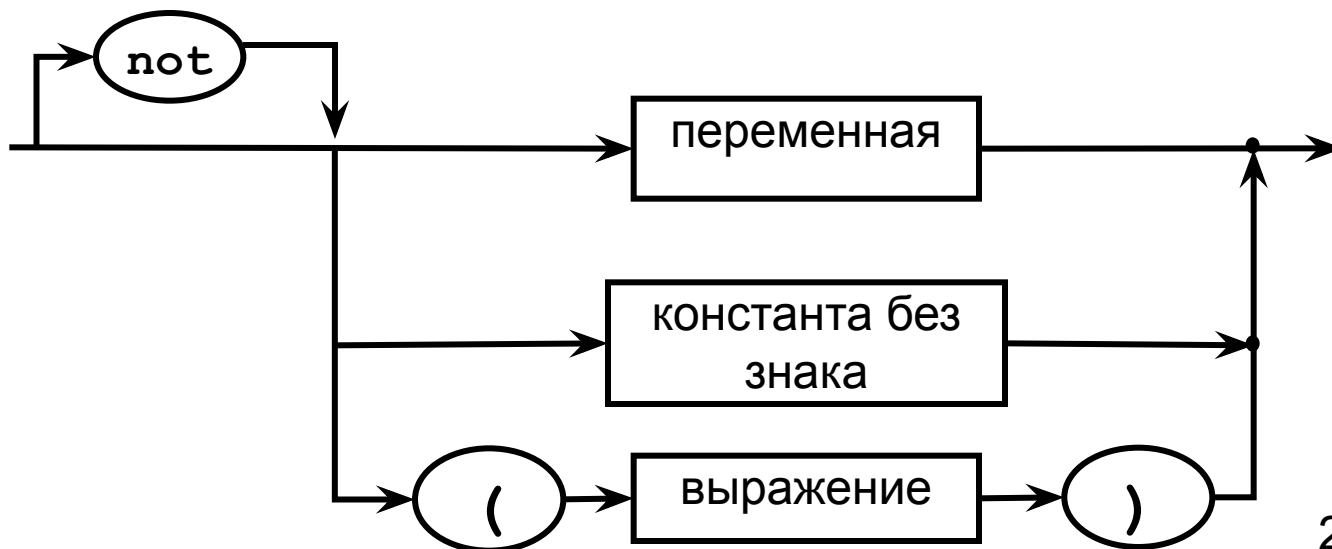
Операции типа сложения: + - **or**

терм



Операции типа умножения: \* / **div mod and**

множитель





## Приоритет (ранги) операций:

операции	ранг
- (унарный минус) <b>not</b>	0
* / <b>div mod and</b>	1
+ - <b>or</b>	2
< <= = <> >= >	3

## Примеры записи арифметических выражений

$$\frac{a1 \cdot x1 - a2 \cdot x2}{x1 - x2} \quad \Longrightarrow \quad ((a1 * x1) - (a2 * x2)) / (x1 - x2)$$

$$a + \left[ w - \frac{m + n(y + x)}{4c} \right] \quad \Longrightarrow \quad a + (w + (m + n * (y + x) / (4 * c)))$$

$$a + b / -c \quad \Longrightarrow \quad a + b / (-c)$$

# Примеры вычисления выражений

```
Var a,b: real; i,j: integer;
```

```
...
```

```
a ← 3.2      b ← 0.1      i ← 1      j ← 2
```

```
a + b ← 3.3      b + i ← 1.1      i + j ← 3
```

```
(a + b) < (b + i) ← False
```

```
a + b < a + b + i ← True
```

```
((a+b<a+b+i) or (j<i)) and ('c'<'a') ← False
```

## 4.5. Стандартные функции

тип аргумента	<b>Integer</b>	<b>Real</b>	<b>Boolean</b>	<b>Char</b>
тип функции				
<b>Integer</b>	Pred Succ Abs Sqr Ins Dec Low High Random SizeOf	Trunc Round SizeOf	SizeOf	Ord SizeOf
<b>Real</b>	Sin Cos Arctan Ln Exp Sqrt	Abs Sqr Sqrt Sin Cos Arctan Ln Exp Int Frac Random		
<b>Boolean</b>	Odd		Low High	
<b>Char</b>	Chr			Pred Succ UpCase

# Примеры задания стандартных функций

Определена как константа величина  $\pi$  - Pi

матем. запись функции	на языке Object Pascal
Sin x	sin(x)
Cos x	cos(x)
Arctg x	arctan(x)
$e^x$	exp(x)
$\sqrt{x}$	sqrt(x)
$x^2$	sqr(x)
x	abs(x)
Ln x	Ln(x)

# Примеры использования стандартных функций

$$\log_{10} b = \frac{\log_e b}{\log_e 10} \quad \Longrightarrow \quad \text{Ln}(b) / \text{ln}(10)$$

$$\sqrt[3]{7} = 7^{1/3} = \exp\left(\frac{1}{3} \ln 7\right) \quad \Longrightarrow \quad \text{Exp}(1/3 * \text{Ln}(7))$$

Задание: запрограммировать формулу:

$$y = \sqrt{\left| \frac{\cos kx - b}{a^2 + b^2} \right|} - e^{|a-b|} + \frac{\text{tg} k^2 x}{a - \sin kx}$$

Исходные данные:  $a = -1.7$ ;  $k = 5$ ;  $x = 5.7$ ;  $b = 2.32$

```
var a,b,x,y:Real;  
    k:integer;
```

```
...
```

```
y:=sqrt(abs((cos(k*x)-b)/(a*a+b*b)))-  
exp(abs(a-b))+sin(k*k*x)/cos(k*k*x)/(a-sin(k*x));
```

# Примеры стандартных функций

`Pred(3) → 2`      `Succ('b') → 'c'`      `Pred(True) → False`

---

`Ord('a') → 97`      `Chr(97) → 'a'`

---

`Odd → true` x-нечет.  
    `→ false` x-чет.

---

`Trunc(3.56) → 3`      `Round(3.56) → 4`

---

`Frac(3.56) → 0.56`

---

`Int(3.56) → 3.0` (ближайшее меньшее число)

`Int(-1.2) → -2.0`

`Int(-3.4) → -4.0`

---

`Inc(x{,n}) → x+n`

(если n = единице шага изменения, то обращение - `Inc(x)`)

`Inc('a') → 'b'`

---

`Dec(x{,n}) → x-n`

(если n = единице шага изменения, то обращение - `Dec(x)`)

`Dec('f') → 'e'`

---

`UpCase('n') → 'N'` (только для латиницы)